Internet Working Group K. Pentikousis, Ed.
Internet Draft EICT
Intended status:
Informational D.
Zhang
Expires: July
2015
Alibaba

January 28, 2015

SUPA Configuration and Policy Mapping
draft-pentikousis-supa-mapping-01

Abstract

Nowadays, the underlying network infrastructure grows in scale and complexity, which make it challenging for network operators to manage and configure the network. Deploying policy or configuration based on an abstract view of the underlying network is much better than manipulating each individual network element, however, in this case, the policy and configuration cannot be recognized by the network elements. This document describes guidelines for mapping configuration and policy into device-level configuration and the way in which such SUPA models will be processed by software to produce configuration details for actual devices. The SUPA framework overview and primary procedures of mapping are proposed. Moreover, an exemplary mapping scenario is provided to illustrate the mechanism involved.

This Internet-Draft will expire on July 28, 2014.

Table of Contents

1. Introduction

As the underlying network infrastructure grows, and new services and traffic are rapidly increased, it becomes significantly more challenging than in the past to maintain the network and deploy new services. Configuration automation can provide significant benefits in deployment agility. Shared Unified Policy Automation (SUPA) [draft-zhou-supa-framework-00] attempts to achieve this configuration automation by introducing multi-level abstractions. In SUPA, the definition of a standardized model for a network topology graph, which could be used to describe topologies at any functional layer,

and information model of various network services and network service development policies allow the network operators to manipulate the network infrastructure as a whole rather than individual devices. Well-designed abstractions are able to provide a wide range of granularity for various applications needs, from the lower-level physical network to high-level network services. However, these information models cannot be directly utilized by network elements, thus a mapping mechanism is necessary to bridge the gap between these information models and network element-recognized configuration.

SUPA employs Management Agent (MA) blocks. MA represents one or more entities that are able to control the operation and management of a network infrastructure, it is utilized between the Operation and Management Application (OAMA) and the network elements to provide , maintain and deploy network services and policies. MA supports the SUPA interface/protocol and is a software repository, which stores the information associated with each network element. The mapping mechanism could be part of MA to help MA to map the SUPA models, into protocol specified configuration models (or so-called southbound interfaces), which is able to be recognized by the network elements.

2. Terminology

This document uses the following terms.

Management Agent (MA): represents one or more entities that are able to control the operation and management of a network infrastructure

Network element (NE): a physical or virtual entity that can be locally managed and operated.

Operation and Management Application (OAMA): represents one or more network entities that are running and controlling network services

SUPA: Shared Unified Policy Automation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Configuration and Policy Mapping

This section introduces a framework for mapping configuration and policy in the context of a network with several network elements and one or more network service systems.

3.1. Overview

The SUPA framework for mapping network-level configuration into specific network management and controlling policies is illustrated in Figure 1. It consists of i) OAMA, ii) MA and iii) NEs.

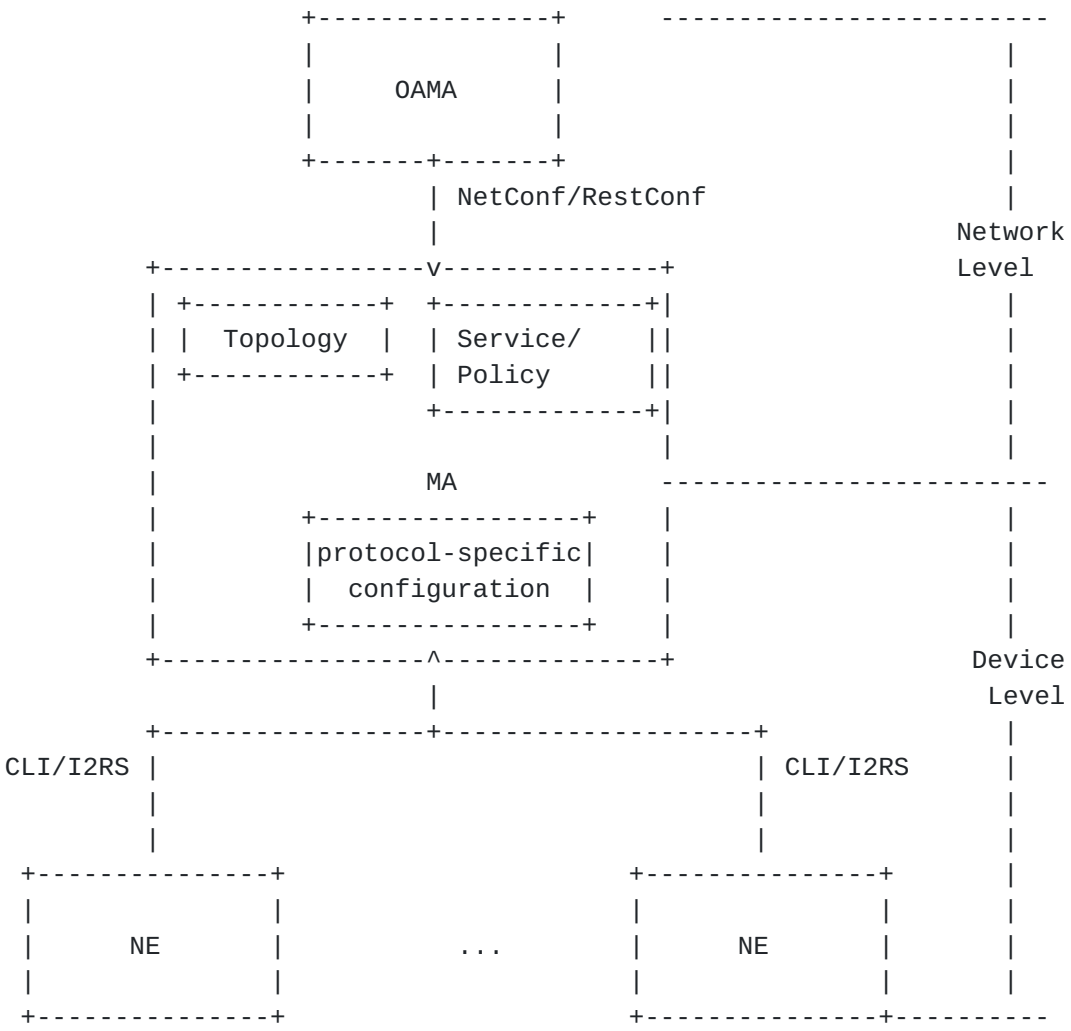


Figure 1: SUPA configuration and policy mapping overview

OAMAOAMA manages and programs the underlying network elements indirectly based on the abstract view of the network infrastructure. In practice, this means that the OAMA can, among others, configure the underlying network as a whole rather than as a set of individual network elements. As a result the diversity of the actual network elements in active operation is abstracted, which allows OAMAOAMA to manage and program the network in a simpler, more maintainable and efficient way. On the other end of the spectrum, the network elements can continue regular operation without having to become cognizant of the fact that configuration is applied at the network level.

In order to bridge the gap between configuration from the OAMA and network elements, the MA has to provide a mapping mechanism which translates the configuration settings from network level to the device level. This document considers three modules in the network management and control system to support such a mapping mechanism, as follows.

First, a topology module maintains the topology of the network infrastructure and provides topology information in the specific network layer as the network service expects. It also provides the necessary information of each network element when mapping configuration from the network-level to device-level. Second, the application/policy configuration module receives the network-level configuration and acts as the primary input of the mapping mechanism. Third, the device configuration produces the output of the mapping mechanism and is responsible for distributing the device-level configuration to the corresponding network elements.

In this framework, one would expect the introduction and use of algorithms/strategies for specific network services which can automatically generate device-level configuration based on the OAMAOAMA policies/configurations. Note, however, that said algorithms/strategies are out of the scope of this document.

3.2. Mapping Procedure

From the view of the OAMA:

Firstly, OAMA needs some context of the underlying network, especially the infrastructure (physical or logical) of the network, before it deploys a policy/service to the network. For example, if OAMA attempts to steer traffic from a path to another, it should have the information of the existing paths first. Otherwise, OAMA maybe

steer traffic to a non-existing path whose links cannot be established practically. OAMA request this context information from MA, and the information is provided with the topology model. This procedure doesn't have to be processed every time OAMA deploys a policy/service.

Secondly, OAMA maybe attempt to get the current status of a policy/service for reference before it deploys a new one. OAMA send a GET request to the MA, and the MA encapsulates this information with the models specified by SUPA network service models or policy models (?).

Thirdly, OAMA deploy a policy/service by sending a "POST" request to the controller with the policy/service information formatted with SUPA models.

From the view of the MA:

Firstly, the MA is responsible for maintaining the infrastructure information, and it provides these information to OAMAs with the topology information model.

Secondly, once the MA receives policy/service models from OAMAs, it maps these models to protocol-specific models. The intelligence/algorithms of how to mapping is out of the scope, and the protocol-specific models is also out of the scope of SUPA. Here, we assume there is a southbound interface - protocol-specific models, however, SUPA doesn't depend on it, the intelligence/algorithms could also translate policy/service models to device-recognized configuration directly as well.

Thirdly, with the protocol-specific models, the device-level configurations for heterogeneous devices can be generated, such as [RFC6020], [RESTCONF], [I-D.ietf-i2rs-architecture] and CLI (Command Line Interface), and the MA distributes these configurations to the corresponding network elements.

3.3. SUPA Mapping Example

Figure 2 illustrates a simple example in which interoperability between OAMA and MA in an inter-data center (inter-DC) environment is considered.

For the purposes of this example, let us focus on the dynamic configuration of the IP path between the seven illustrated DCs,

labeled A, B, C, D, E, F and G, based on the policies. First of all, we would like the IP path to be created based on certain constraints. Secondly, we would like to map it to the device-level connections. In this scenario, there are two paths from DC A to DC B. Typical IP shortest-path routing would choose path A(1.1.1.1)-C(3.3.3.3)>B(2.2.2.2). However, under certain conditions, such as, for instance, when the bandwidth between A and B is not suitable, the NSS can decide that is better to steer traffic from path (A, C, B) to path (A, D, E, B).

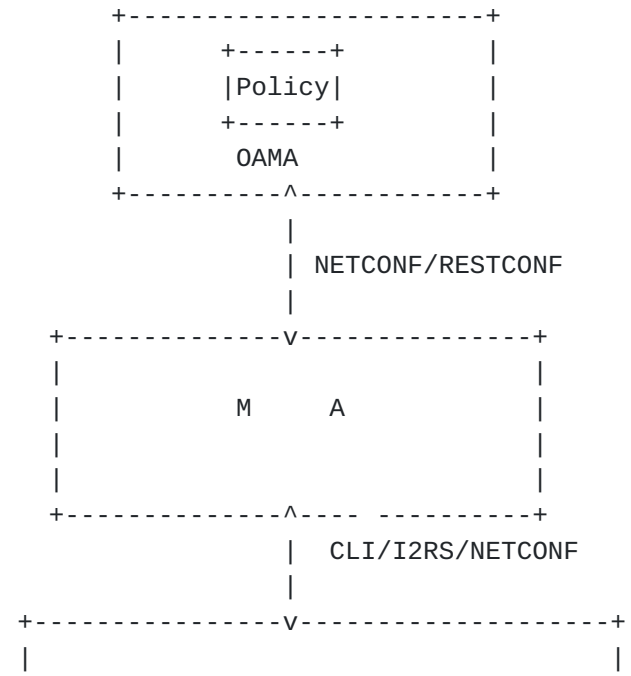
Figure 2 depicts the layer 3 topology of the underlying network.. At first, OAMA needs some information about A, B, C, D and the links between them. This information can be obtained from OM, and it is listed as below. It should be noted that some nodes and links are skipped because of the limited space. This information is derived from the Topology YANG model described in [draft-contreras-supa-yang-network-topo-02].

```
<topologies>
 <topology>
   <topoId>1111111100000000</topoId>
   <topoName>mapping_topo</topoName>
   <layer>ip</layer>
 </topology>
 <nodes>
   <node>
     <nodeID>1.1.1.1</nodeID>
     <nodeName>A</nodeName>
     <nodeType>physical</nodeType>
     <adminStatus>adminUp</adminStatus>
     <operStatus>up</operStatus>
     <parentTopoID>1111111100000000</parentTopoID>
   </node>
   <node>
     <nodeID>2.2.2.2</nodeID>
     <nodeName>B</nodeName>
     <nodeType>physical</nodeType>
     <adminStatus>adminUp</adminStatus>
     <operStatus>up</operStatus>
     <parentTopoID>1111111100000000</parentTopoID>
   </node>
          ......
   <node>
     <nodeID>3.3.3.3</nodeID>
     <nodeName>C</nodeName>
     <nodeType>physical</nodeType>
     <adminStatus>adminUp</adminStatus>
     <operStatus>up</operStatus>
     <parentTopoID>1111111100000000</parentTopoID>
   </node>
 </nodes>
 <links>
   <link>
     <linkId>1</linkId>
     <linkName>A2C</linkName>
     <linkType>telink</linkType>
     <direction>bidirectional</direction>
     <adminStatus>adminUp</adminStatus>
     <operStatus>up</operStatus>
     <sourceNodeId>1.1.1.1</sourceNodeId>
     <destinationNodeId>3.3.3.3</destinationNodeId>
     <parentTopoID>1111111100000000<parentTopoID>
     <linkTeAttrCfg>
       <maxReservableBandwidth>2000</maxReservableBandwidth>
     </linkTeAttrCfg>
```

```
        </link>
                .....

        <link>
          <linkId>2</linkId>
          <linkName>C2B</linkName>
          <linkType>telink</linkType>
          <direction>bidirectional</direction>
          <adminStatus>adminUp</adminStatus>
          <operStatus>up</operStatus>
          <sourceNodeId>3.3.3.3</sourceNodeId>
          <destinationNodeId>2</destinationNodeId>
          <parentTopoID>1111111100000000<parentTopoID>
          <linkTeAttrCfg>
            <maxReservableBandwidth>50000</maxReservableBandwidth>
          </linkTeAttrCfg>
        </link>
      </links>
    </topologies>
```

Secondly, the OAMA sends the steering information to MA using a protocol such as NETCONF or RESTCONF.



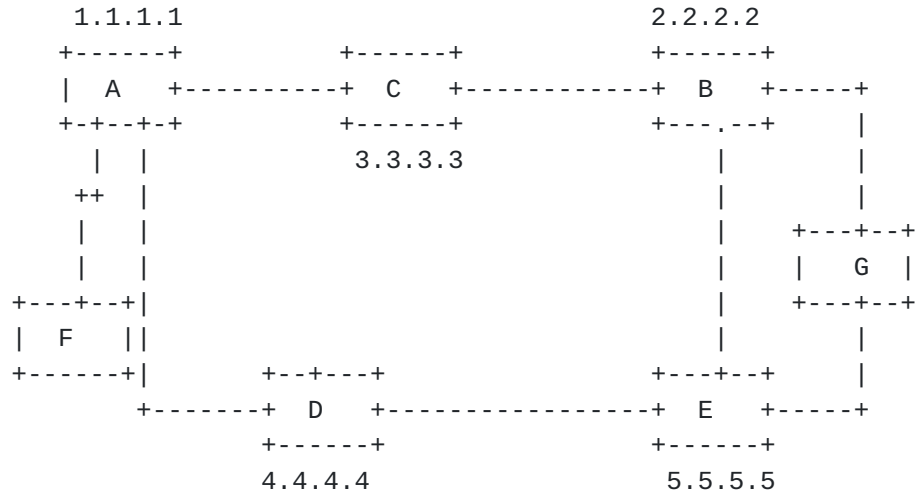


Figure 2: Bandwidth usage optimization for DC Interconnection

Figure 3 presents the requirements for traffic steering: the traffic (supa_flow) whose destination IP address is 11.11.11.11/24 needs to be steered to DC B, the new path must go through DC D. This configuration is derived from the YANG model described in [draft-xxx-supa-configuration-model-00].

```
<specifyFlowPaths>
 <vpnName>supa_vpn</vpnName>
 <vpnType>L3VPN</vpnType>
 <flowName>supa_flow</flowName>
 <node>4.4.4.4</node>
</specifyFlowPaths>
```

Figure 3: Example traffic steering requirements

Based on this configuration, the MA generates a path which meets the requirements, in this example, the computed path is (A, D, E, B). MA also has to configure each device on the new path, not only the devices specified by the configuration such as node D, but also the devices in the underlying network which must be reconfigured, such as node E. The topology information is also necessary when MA decides which device ought to be configured.

With the assistance of other information in MA, such as topology information, service/policy configuration can be translated into protocol-specific yang models (or southbound interface) first. Taking

node D as an example, the configuration could be as follows when Yang models defined in [I-D.ietf-netmod-routing-cfg] is utilized.

```
<rt:routing>
<rt:routing-instance>
  <rt:name>rtr0</rt:name>
  <rt:description>Router D</rt:description>
  <rt:routing-protocols>
    <rt:routing-protocol>
      <rt:type>rt:static</rt:type>
      <rt:name>st0</rt:name>
      <rt:description>
        Static routing is used for the internal network.
      </rt:description>
      <rt:static-routes>
        <v4ur:ipv4>
          <v4ur:route>
            <v4ur:destination-prefix>
              11.11.11.11/24
            </v4ur:destination-prefix>
            <v4ur:next-hop>
              <v4ur:next-hop-address>
                5.5.5.5
              </v4ur:next-hop-address>
            </v4ur:next-hop>
          </v4ur:route>
        </v4ur:ipv4>
      </rt:static-routes>
    </rt:routing-protocol>
  </rt:routing-protocols>
 </rt:routing-instance>
</rt:routing>
```

The configurations of other nodes are not listed because of the limited space. Once nodes A, C, D and E have received their respective protocol-specific configurations, the device-level configuration could be deployed and then, the traffic is steered as OAMA expects.

4. Security Considerations

Security considerations will be discussed in an upcoming revision of this document.

5. IANA Considerations

TBD

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[draft-adel-supa-configuration-model-00] Adel Zaalouk, K.Pentikousis, W. Liu, "A YANG Data Model for Configuration of SUPA (Shared Unified Policy Automation)" (work inprogress), September 2014.

[draft-zhou-supa-framwork-00] C. Zhou, D.Lopez, G.Karagiannis and Q.Sun "The Architecture for Shared Unified Policy Automation (SUPA)", draft-zhou-supa-architecture-00, (work inprogress), September 2014.

[I-D.ietf-i2rs-architecture] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the RoutingSystem", draft-ietf-i2rs-architecture-04 (work inprogress), June 2014.

[I-D.ietf-netmod-routing-cfg] Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-15 (work in progress), May 2014.

[I-D.hares-i2rs-info-model-policy] Hares, S. and W. Wu, "An Information Model for Networkpolicy", draft-hares-i2rs-info-model-policy-02 (work inprogress), March 2014.

[RESTCONF] Bierman, A., Bjorklund, M., Watsen, K., and R. Fernando, "RESTCONF Protocol", draft-ietf-netconf-restconf-01 (workin progress), July 2014.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

7. Acknowledgments

This document has benefited comments, suggestions, and proposed text provided by Cathy Zhou and Will Liu (listed in alphabetical order).

Junru Lin and Zhayiyong contributed to an earlier version of this draft.

Authors' Addresses

Kostas Pentikousis (editor)
EICT GmbH
Torgauer Strasse 12-15
Berlin 10829
Germany
Email: k.pentikousis@eict.de

Dacheng Zhang
Alibaba
Chaoyang Dist
Beijing 100000
P.R. China
Dacheng.zdc@alibaba-inc.com